Loss Probabilities for Circuit-Switched Networks

SA Berezner
Dept of Mathematical Statistics
University of Natal
4001 Durban South Africa
E-mail: berezner@ph.und.ac.za
Facsimile: +27 31 2616550

AE Krzesinski
Dept of Computer Science
University of Stellenbosch
7600 Stellenbosch South Africa
E-mail: aek1@cs.sun.ac.za
Facsimile: +27 21 8084416

Abstract: Consider a circuit-switched network where several source switches are connected to a destination switch via a tandem switch. Circuit-switched networks traditionally employ a Blocked Calls Cleared (BCC) admission rule: a call is rejected if all circuits from the originating switch to the tandem switch are busy, or if all circuits from the tandem switch to the destination switch are busy. This paper investigates a simple extension to the BCC rule. Rather than reject all blocked calls, the Blocked Calls Held (BCH) rule holds some blocked calls by storing their call signalling information in a buffer at the tandem switch. These calls will later be connected when circuits become available. The BCH system is modeled by a MSCCC loss queue. The stationary distribution for the MSCCC loss queue is presented and an efficient recursive calculation of the queue performance measures is derived. The model reveals that, under moderate overload conditions, the BCH rule achieves a substantially lower loss probability than the BCC rule, at the expense of a small increase in the connection delay.

Keywords: loss queues, MSCCC queues, order independent queues, product form networks.

Computing Review Category: C.4 Computer Systems Organization: Performance of Systems. D.4.8 Operating Systems: Performance.

1. INTRODUCTION

Consider figure 1 which represents (part of) a circuit-switched network. There are C source switches $1 \dots C$. Each switch c is connected by an access link c consisting of B_c circuits to a tandem switch $\mathcal{S}$ which switches the incoming calls among B outgoing circuits to a switch $\mathcal{D}$. In general $B < B_1 + \dots + B_C$. The case $B \geq B_1 + \dots + B_C$ is both unnatural and mathematically trivial.

Calls arrive at switch c according to a Poisson process with rate λ_c. A call from switch c will be connected to the destination switch $\mathcal{D}$ if a circuit from link c and a circuit from $\mathcal{S}$ to $\mathcal{D}$ are available. Both circuits are held for the duration of the call. Call holding times are exponentially distributed with mean $1/\mu$. Both circuits are released when the call is cleared. We investigate two models of call connection which are distinguished by having different call admission rules (1) Blocked Calls Cleared (BCC): if the connection request finds that all circuits from $\mathcal{S}$ to $\mathcal{D}$ are in use, or all B_c circuits in link c are busy, then the call is lost (2) Blocked Calls Held (BCH): if the connection request finds that all circuits from $\mathcal{S}$ to $\mathcal{D}$ are in use, or all B_c circuits in link c are busy, then the call will be held (blocked, queued) if the total number of calls in the system (blocked and connected) is less than N or the call will be lost if the total number of calls in the system (blocked and connected) is equal to N where $N > B$.

[1]This work is supported by a grant from the Foundation for Research Development Communications Systems Programme

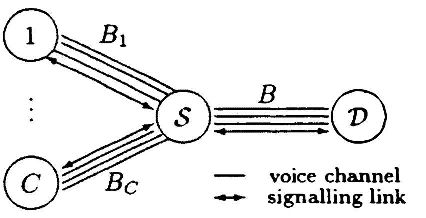

Figure 1: Switching network with signalling

The capability for storing blocked calls is provided by a common channel signalling facility which provides a dedicated signalling link among the switches. The signalling link is separate from the transmission links and is shared among all the transmission links. With reference to figure 1, signalling information for each call is kept in a queue at the tandem switch $\mathcal{S}$ until the call is cleared. When a call completes, the system will attempt to connect the call with the earliest arrival time. The queue is scanned from the front looking for the first call that can be connected. If the call (say it originated at switch c) cannot be connected because all B_c circuits in link c are busy, the next call in the queue is considered. Connection requests are thus attempted on a FCFS basis. The Multiserver Centre with Concurrent Classes of Customers (MSCCC) [4, 7] provides an exact analytical solution to the BCH model.

In both models the quantities of interest are the loss and the blocking probabilities and the blocking (connection) delays. The BCC model has a simple analytic solution and efficient numerical methods have been developed [5, 8] to compute exact and asymptotic solutions for large networks of loss queues. This paper presents an efficient computational method for computing the performance measures of the MSCCC loss queue which is used to model the BCH system.

Note that for practical purposes, decreasing a loss probability, say equal to 10^{-2}, by a factor of two is worth doing only if it can be achieved by simple call management procedures and if it does not result in a significant connection delay. As we show below, the BCH admission rule meets both these conditions.

The amount of storage required to compute the loss probability for a MSCCC queue is of order NCB^2. For large models it therefore interesting to investigate to what extent the $M/M/B/N$ queue provides a useful approximation to the MSCCC loss queue where it is assumed that calls are blocked due to an under-provision of circuits at the tandem switch $\mathcal{S}$ rather than because of a lack of circuits on the access links.

2. ORDER INDEPENDENT QUEUES

Consider a queue serving customers of type c where $c \in \mathcal{C}$ where $\mathcal{C}$ is a finite set. Customers of type c arrive individually at the instants of a Poisson stream with rate λ_c. The customers, whether waiting or in service, form a queue in the order of their arrival. Arriving customers join the back of the queue and the front of the queue is identified with position 1. Each customer of type c presents a demand for service time which is exponentially distributed with mean $1/\mu$. All the random variables involved in the description of the queue are independent.

Let $\vec{C} = (c_n \ldots c_1)$ denote the state of a queue of length n where c_i denotes the type of the customer in position i, $i = 1 \ldots n$. Let 0 denote the empty queue and $\mathcal{S} = \{0\} \cup \bigcup_{n=1}^{\infty} \mathcal{C}^n$ denote the state space of the queue.

Let the total service effort in state $(c_n \ldots c_1)$ be supplied at the rate $\phi(c_n \ldots c_1)$. A portion $\gamma_i(c_n \ldots c_1)$ of the total service effort is directed at the customer in queue position i, $i = 1 \ldots n$. When the customer in queue position i completes service, the customer departs and the gap in the queue is closed by the obvious shift: the customers in positions $i+1, i+2, \ldots, n$ move to positions $i, i+1, \ldots, n-1$ respectively. We do not require that $\sum_{i=1}^{n} \gamma_i(c_n \ldots c_1) = 1$ so that a part of the service facility might be wasted. This is an important and distinguishing feature of the queue.

We next impose certain conditions on the functions ϕ and γ. Firstly we require that in any state $(c_n \ldots c_1) \in \mathcal{S}$ the rate at which departures (service completions) occur is independent of the order of the customers in the queue and is thus the same for any state $(c_{\sigma(n)} \ldots c_{\sigma(1)})$ where σ denotes any permutation of $(1 \ldots n)$. Secondly we require that the (relative) proportion of the service effort supplied to the customer in queue position i depends only on the composition of the queue up to and including position i. This implies that when a server becomes free the queue is searched from the front to the back for a customer which can be admitted into service. Lastly we assume that in any state (except for the empty queue) there is a positive rate of service completion. This condition is required in order to ensure the irreducibility of the Markov chain.

A formal description of the three conditions presented above can be given as follows. Consider the queue in state $(c_n \ldots c_1)$. The departure rate of the customer in queue position i is given by $\phi(c_n \ldots c_1)\mu_{c_i}\gamma_i(c_n \ldots c_1)$ and the total departure rate is given by the sum of these quantities over all the positions in the queue.

The queue is said to be an Order Independent (OI) queue if, for all $(c_n \ldots c_1) \in \mathcal{S}$ and all $i = 1 \ldots n$ the rates of service completion can be factored as

$$\phi(c_n \ldots c_1)\mu_{c_i}\gamma_i(c_n \ldots c_1) = \mu(n) s_i(c_n \ldots c_1)$$

such that

(i) $s_i(c_n \ldots c_1) = s_i(c_i \ldots c_1)$ for $1 \leq i \leq n$

(ii) $k(c_n \ldots c_1) = \sum_{i=1}^{n} s_i(c_n \ldots c_1)$ is independent of permutations of $(c_n \ldots c_1)$

(iii) $\mu(n) > 0$ for $n > 0$ and $s_1(c) > 0$ for any $c \in \mathcal{C}$

3. ANALYSIS OF THE QUEUE

The restrictions (i)–(iii) on $s_i(\cdot)$ are sufficient to ensure that the OI queue is quasi-reversible at equilibrium and that the stationary distribution exists and is given by

$$\pi(c_n \ldots c_1) = b \prod_{i=1}^{n} \frac{\lambda_{c_i}}{\mu(i) k(c_i \ldots c_1)} \tag{1}$$

if and only if the normalising constant b is finite. The proof is given in [2].

Equation (1) is too detailed to be of practical use when computing the performance measures of the OI queue. In order to reduce the complexity of these equations, we use the fact that $k(c_n \ldots c_1)$ is independent of the order of the customers in $(c_n \ldots c_1)$.

Let M_c denote the number of type c customers in $\vec{C}$. Let $\pi(\vec{M})$ denote the probability that the queue is in state $\vec{M} = (M_1 \ldots M_C)$. Let $k(\vec{M})$ denote the total number of customers in service when the queue is in state $\vec{M}$. It can be shown [2] that

$$\mu(|\vec{M}|) k(\vec{M}) \pi(\vec{M}) = \sum_{c \in \mathcal{C}} \lambda_c \pi(\vec{M} - \vec{1}_c) \tag{2}$$

The recursive equation (2) can be further simplified if the service effort directed at a specific customer type can be determined. Define

$$k_c(c_n \ldots c_1) = \sum_{i=1}^{n} s_i(c_n \ldots c_1) 1(c_i = c).$$

Although condition (ii) requires that $k(\vec{C})$ is independent of permutations of $\vec{C}$, this does not necessarily imply that $k_c(\vec{C})$ is independent of permutations of $\vec{C}$. However, in those cases where $k_c(\vec{C})$ is independent of permutations of $\vec{C}$ define

$$k_c(\vec{M}) = k_c(\vec{C})$$

It can be shown [2] that equation (2) reduces to

$$\mu(|\vec{M}|)k_c(\vec{M})\pi(\vec{M}) = \lambda_c\pi(\vec{M} - \vec{1}_c)$$

which provides a recursion for the efficient computation of the aggregated probabilities and the performance measures of the queue.

4. THE OI LOSS QUEUE

The OI loss queue places a restriction on the maximum number N of customers in the queue. Customers arriving to a queue which contains N customers are lost. The state space of the queue is $\mathcal{S}_N = \{0\} \cup \bigcup_{n=1}^{N} \mathcal{C}^n$. For any $c \in \mathcal{C}, n < N$ and any $(c_n \ldots c_1) \in \mathcal{S}$, the transition rate due to type c arrivals is λ_c and there are no arrivals to the states in $\mathcal{C}^N$. The latter restriction implies that the OI loss queue is not quasi-reversible. However, the stationary distribution for the loss queue can be obtained in the same way as the distribution of the OI queue and is given by [2]

$$\pi(c_n \ldots c_1) = b \prod_{i=1}^{n} \frac{\lambda_{c_i}}{\mu(i)k(c_i \ldots c_1)} \tag{3}$$

5. MSCCC LOSS QUEUE: PERFORMANCE MEASURES

An efficient calculation of the performance measures is obtained by grouping probabilities over different sets. Recursive expressions for the performance measures are expressed as multiples of the starting value of the recursion – the (unknown) probability $\pi_N(0)$ of the empty queue. We define the unnormalized (proportional) probabilities $P_N(\cdot) = \pi_N(\cdot)/\pi_N(0)$ which satisfy the same recursive relations as $\pi_N(\cdot)$. However, in this case the starting value $P_N(0)$ is equal to unity. The value of $\pi_N(0)$ can then be calculated as the reciprocal of the sum of the unnormalized probabilities $P_N(\cdot)$.

Consider the same MSCCC loss queue with two different limits N_1 and N_2 on the total number of customers where $N_1 \leq N_2$. The unnormalized probabilities $P_{N_1}(\cdot)$ and $P_{N_2}(\cdot)$ coincide on the set $\mathcal{S}_N$ since they satisfy the same recursive equations and have the same starting value (unity) in the recursion. We can therefore write $P(\cdot)$ instead of $P_N(\cdot)$ – an important observation that is used in the derivation of $\pi_N(0)$.

5.1. Probabilities Over Special Sets

This section and the following section presents the main results for the calculation of the performance measures for a MSCCC loss queue. The details of the analysis can be found in [3]. Where no confusion can arise we omit the subscript N.

In the domain $b < B$ where not all servers are busy, the number of type c customers in service is given by $\min(M_c, B_c)$. Let $P(n, b, c, i)$ denote the probability that n customers are present, b servers are busy, i customers of type c are present and no customers of types $c + 1 \ldots C$ are present. For $1 \leq c \leq C$, $1 \leq i \leq N$, $i \leq n \leq N$ and $\min(i, B_c) \leq b < B$

$$P(n,b,c,i) \;=\; \begin{cases} \frac{\rho_c}{i} P(n-1, b-1, c, i-1) & 0 < i \leq B_c \\ \frac{\rho_c}{B_c} P(n-1, b, c, i-1) & B_c < i \leq n \end{cases}$$

For $1 \leq c \leq C$, $B_c \leq b < B$ and $b < n \leq N$ define

$$\begin{aligned} P^+(n,b,c) &= \sum_{i=B_c+1}^{n} P(n,b,c,i) \\ &= \frac{\rho_c^{B_c+1}}{B_c! B_c} P(n-1-B_c, b-B_c, c-1) + \frac{\rho_c}{B_c} P^+(n-1, b, c) \end{aligned}$$

Let $P(n,b,c)$ denote the probability that n customers are present, b servers are busy and no customers of types $c+1 \ldots C$ are present. For $1 \leq c \leq C$, $0 \leq b < B$ and $b \leq n \leq N$

$$P(n,b,c) \;=\; P^+(n,b,c) + \sum_{i=0}^{\min(b,B_c)} \frac{\rho_c^i}{i!} P(n-i, b-i, c-1)$$

where $P(n,b,c,0) = P(n,b,c-1)$. Successive applications of equation (4) with the initial values

$$P(n,b,c) = \begin{cases} P(0) = 1 & n = b = 0 \\ 0 & \text{otherwise} \end{cases}$$

for $0 \leq c \leq C$ yield values for the probability $P(n,b)$ that n customers are present and b servers are busy.

Let $P(n)$ denote the probability that n customers are present. For $0 \leq n < B$

$$P(n) = \sum_{b=0}^{n} P(n,b)$$

To calculate $P(n)$ for $B \leq n \leq N$ note that the arrival rate of accepted traffic (the accepted traffic is the offered traffic minus the lost traffic) is equal to the departure rate of customers from the queue. Let $\lambda = \lambda_1 + \cdots + \lambda_C$. For $B \leq n \leq N$

$$\lambda \sum_{j=0}^{n-1} P_n(j) \;=\; \mu \sum_{b=1}^{B-1} b \sum_{j=b}^{n} P_n(j,b) + \mu B \sum_{j=B}^{n} P_n(j,B)$$

As mentioned previously, the unnormalized probabilities are not dependent on the subscript n, so the above equation becomes

$$\rho \sum_{j=0}^{n-1} P(j) \;=\; \sum_{b=1}^{B-1} b \sum_{j=b}^{n} P(j,b) + B \sum_{j=B}^{n-1} P(j,B) + BP(n,B) \qquad (4)$$

where $\rho = \lambda/\mu$. Equation (4) yields $P(n,B)$ so that $P(n)$ can be computed as

$$P(n) = \sum_{b=0}^{B} P(n,b) \qquad (5)$$

Successive applications of equations (4) and (5) for $B \leq n \leq N$ together with the known starting values $P(0) \ldots P(B-1)$ yield the values of $P(B) \ldots P(N)$. The normalized probabilities $\pi_N(n)$ for $0 \leq n \leq N$ are given by

$$\pi_N(n) = \pi_N(0) P(n)$$

where

$$\pi_N(0) = 1 / \sum_{n=0}^{N} P(n)$$

5.2. Average Queue Lengths

Let $L(n, b)$ denote the average number of type C customers at the queue calculated at instants when n customers are present and $b < B$ servers are busy. For $1 \leq b < B$ and $b \leq n \leq N$

$$L(n,b) = \sum_{i=1}^{n} i\pi(n,b,C,i)$$

For $B_C \leq b < B$ and $b < n \leq N$ define

$$L^+(n,b) = \sum_{i=B_C+1}^{n} i\pi(n,b,C,i)$$

It can be shown [3] that

$$\frac{B_C}{\rho_C} L^+(n,b) = F(c)\pi(n-1-B_C, b-B_C, C-1) + L^+(n-1,b,C) + \pi^+(n-1,b,C)$$

where $F(c) = (B_C+1)\rho_C^{B_C}/B_C!$

For $1 \leq b < B$ and $b \leq n \leq N$

$$\begin{aligned} L(n,b) - L^+(n,b) &= \sum_{i=1}^{\min(b,B_C)} i\pi(n,b,C,i) \\ &= \sum_{i=1}^{\min(b,B_C)} \frac{i\rho_C^i}{i!}\pi(n-i,b-i,C-1) \end{aligned}$$

Let $L(n)$ denote the average number of type C customers at the queue calculated at instants when n customers present. For $1 \leq n < B$

$$L(n) = \sum_{b=1}^{n} L(n,b)$$

To calculate $L(n)$ for $B \leq n \leq N$ write

$$BL(n) = \sum_{b=1}^{B} bL(n,b) + \sum_{b=1}^{B-1}(B-b)L(n,b) \tag{6}$$

It can be shown [3] that

$$\sum_{b=1}^{B} bL(n,b) = \rho L(n-1) + \rho_C\pi(n-1) \tag{7}$$

where $\rho = \sum_{j=1}^{C} \rho_j$ and

$$\pi(n) = \sum_{b=1}^{min(n,B)} \pi(n,b)$$

Equations (6) and (7) yield

$$BL(n) = \rho L(n-1) + \rho_C\pi(n-1) + S(n)$$

for $1 \leq n \leq N$ where

$$S(n) = \sum_{b=1}^{B-1}(B-b)L(n,b)$$

Let L_C denote the average number of type C customers present.

$$L_C = \sum_{n=1}^{N} L(n) \tag{8}$$

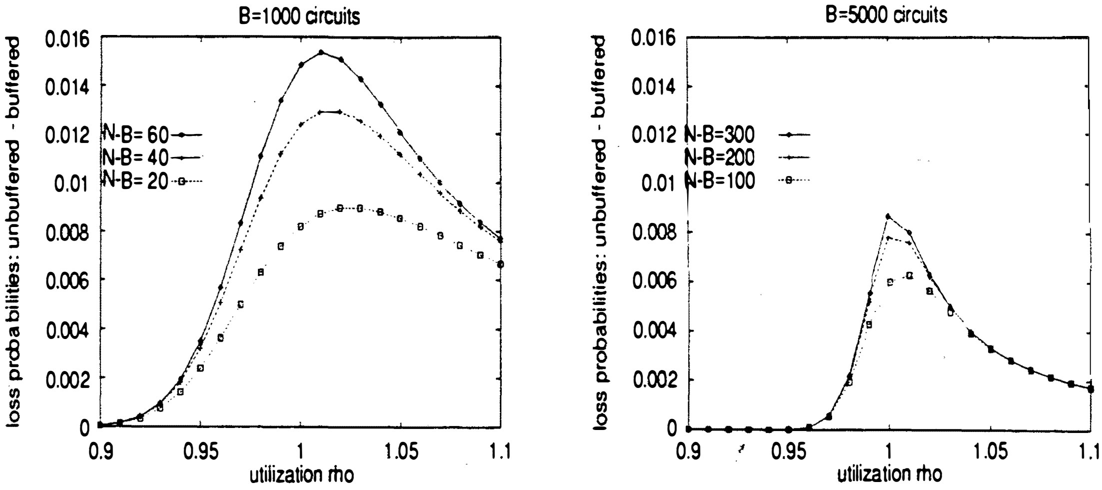

Figure 2: Loss probability $\pi_B(B) - \pi_N(N)$

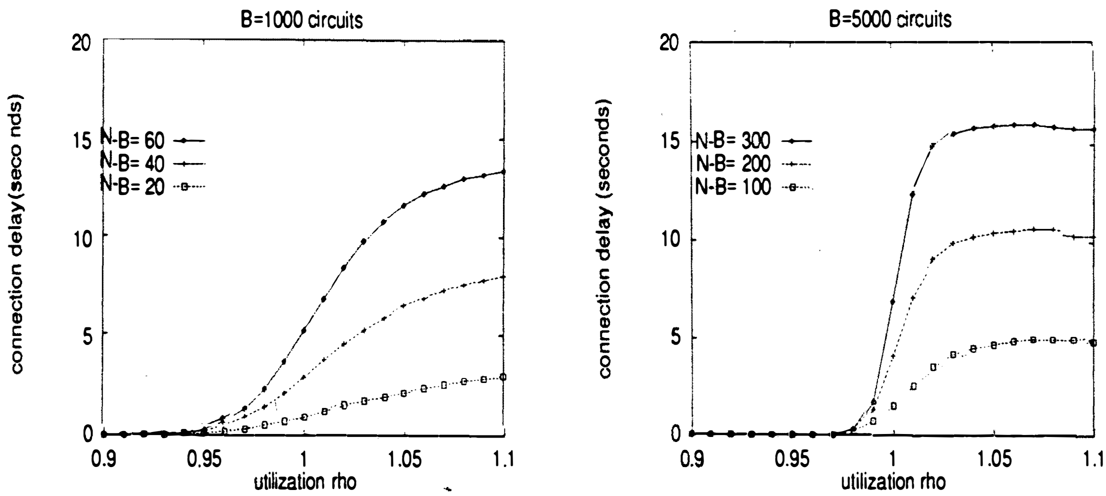

Figure 3: Average connection delay

Equation (8) is applied C times to successive cyclic permutations of the $1 \ldots C$ customer types to yield the average queue lengths L_c for all $c \in \{1 \ldots C\}$.

Summing the values of L_c over all $c \in \mathcal{C}$ yields the expected queue length. This result can be computed at an earlier stage as $L = \sum_{n=1}^{N} n\pi(n)$ - which provides a useful check on the correctness of the computer programme which calculates the values of L_c.

6. AN APPLICATION

This section investigates three models of call loss: BCC; the $M/M/B/N$ queue; and BCH. Figure 2 illustrates [1] the use of the $M/M/B/N$ model to predict the improvement in loss probability $\pi_B(B) - \pi_N(N)$ for several values of the buffer size $N - B$. When the tandem switch is equipped with $B = 1000$ circuits, the addition of even a small buffer sufficient to store 20 calls results in a substantial reduction in the loss probability and the loss probability can be further reduced by increasing the size of the buffer. For a larger model ($B = 5000$ circuits), a buffer of size 100 is sufficient to achieve a substantial reduction in the loss probability - in this case, increasing the buffer size beyond 100 offers little benefit. In each case, the best improvement occurs when the tandem switch is subject to a moderate overload $\rho \sim 1$. Call holding reduces the loss probability at the expense of introducing a connection delay. Figures

ρ	B_c	$\rho_c B_c$	BCC	$M/M/B/N$		BCH	
			loss	delay	loss	delay	loss
1.02	110	102	0.055	3.1	0.034	4.1	0.036
1.00	110	100	0.043	2.3	0.022	3.3	0.024
0.98	110	98	0.032	1.5	0.012	2.4	0.014
0.95	110	95	0.019	0.7	0.004	1.3	0.005
0.90	110	90	0.006	0.07	0.0003	0.3	0.0003

Table 1:

c	B_c	$\rho_c B_c$	BCC	$M/M/B/N$		BCH	
			loss	delay	loss	delay	loss
1	180	170	0.042	2.3	0.022	3.1	0.024
2	130	120	0.043	2.3	0.022	3.3	0.024
3	90	80	0.044	2.3	0.022	3.3	0.024
4	40	30	0.038	2.3	0.022	2.4	0.024

Table 2: $\rho_c < 1$ and $\rho = 1$

2 and 3 show the trade-off involved. In the case of a switch with $B = 5000$ circuits, a large buffer sufficient to store 300 calls reduces the loss probability but the resulting connection delay may be sufficiently large (up to 15 seconds) to discourage subscribers who may clear their calls before being connected. A smaller buffer sufficient to store 100 calls will offer a substantial reduction in the loss probability and introduce an acceptable connection delay of less than 5 seconds.

Tables 1 through 4 compare the loss probabilities and connection delays for the three models. The following parameters were fixed: the number of outbound circuits $B = 400$; the number of source switches $C = 4$; the average holding time $1/\mu = 200$ seconds; and the limit on the total number of calls in the system $N = 420$ except for the BCC model where $N = B = 400$. These values were chosen for clarity of presentation: the computational algorithms are sufficiently efficient to permit models with larger values of B (several 1000's), many access links (several 10's) and large buffer sizes.

Table 1 presents an example where all access links have the same capacity and offer the same traffic. The connection delays and loss probabilities are therefore identical for each access link. The table confirms that when $\rho \sim 1$ call holding results in a substantial reduction in the loss probability at the expense of a small connection delay.

Tables 2 through 4 present examples where the number of circuits B_c and the traffic intensities ρ_c are different for each access link c. Table 2 presents the case where all the $\rho_c < 1$ and $\rho = 1$. The table reveals that call holding (BCH) results in a substantial reduction in the loss probability without a major increase in the connection delay.

Table 3 presents the case where link from switch 4 is overloaded ($\rho_4 = 1.5$) although the total

c	B_c	$\rho_c B_c$	BCC	$M/M/B/N$		BCH	
			loss	delay	loss	delay	loss
1	180	150	0.003	0.7	0.004	0.00	0.333
2	130	100	0.002	0.7	0.004	0.00	0.333
3	90	70	0.004	0.7	0.004	0.00	0.333
4	40	60	0.360	0.7	0.004	823	0.333

Table 3: $\rho_4 > 1$ and $\rho = 0.95$

			BCC	$M/M/B/N$		BCH	
c	B_c	$\rho_c B_c$	loss	delay	loss	delay	loss
1	180	170	0.031	0.7	0.004	4.0	0.006
2	130	110	0.012	0.7	0.004	0.6	0.006
3	90	70	0.010	0.7	0.004	0.5	0.006
4	40	30	0.020	0.7	0.004	1.3	0.006

Table 4: $\rho_c < 1$ and $\rho = 0.95$

traffic intensity $\rho = \sum_{c=1}^{C} \rho_c B_c / B < 1$. The BCC mechanism successfully prevents the calls from switch 4 from overwhelming the system: the calls from switch 4 experience a large loss probability and the high arrival rate from switch 4 has no effect on the loss probabilities on the other access links. The $M/M/B/N$ model is inappropriate under these circumstances: the model assumes that information concerning the origin of the calls is lost so the model does not represent the selective rejection applied to calls from switch 4. The BCH model does describes the interplay among the calls from the various switches. However, the model reveals that call holding under these circumstances ($\rho_4 > 1$) leads to unacceptably high loss probabilities for all calls. Unlike the BCC model, the BCH model does not selectively reject calls from switch 4, so the queue becomes congested with blocked calls from switch 4 even though these calls provide only a modest share (16 percent) of the total arrival traffic. Calls from the other switches are lost even though not all B outbound circuits are busy.

Table 4 presents the case when all $\rho_c < 1$ and $\rho < 1$. The table shows that under these circumstances the loss probabilities can be substantially reduced by adding a small buffer at the expense of a small connection delay. The fact that the loss probabilities are the same for all access links implies that BCH is a fair (unbiased) admission mechanism – which might or might not be desirable.

Tables 1 through 4 show that in those cases where B is large and $\rho \sim 1$ so that losses are negligible on the access links, the $M/M/B/N$ queue offers a good approximation for the MSCCC loss queue (BCH). However, this is not true in general – the $M/M/B/N$ queue cannot model the type dependent blocking constraints which are a distinguishing feature of the MSCCC queue. Thus if we consider systems whose behaviour is dominated by blocking on the access links, then the difference between the $M/M/B/N$ and the MSCCC loss queues becomes noticeable.

7. CONCLUSION

The provision of a separate signalling channel in a circuit-switched network allows signalling information for blocked calls to be stored in a buffer until circuit(s) become free to accept the call. This call queueing mechanism can be modeled by a MSCCC loss queue. The model was used to investigate the effect of buffer size on the loss probabilities and it was shown that the

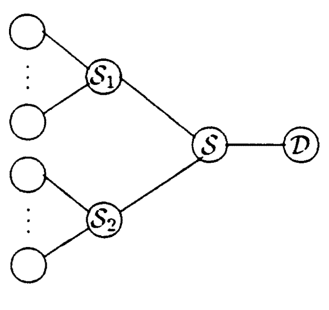

Figure 4: Hierarchical network

use of a small buffer significantly reduces the loss probability at the cost of introducing a small connection delay.

The network presented in figure 1 can be generalized [8] to an hierarchical loss network presented in figure 4. Consider two circuit-switching networks with tandem switches $\mathcal{S}_1$ and $\mathcal{S}_2$ connected to another tandem switch $\mathcal{S}$ which in turn is connected to a destination switch $\mathcal{D}$. This network, as well as networks with a higher degree of hierarchical ordering (each switch can home on one hierarchically superior switch) can be modeled as a Multiserver Station with Hierarchical Concurrency Constraints (MSHCC) [6].

References

[1] S.A. Berezner and A.E. Krzesinski, Blocking Probabilities in Large Circuit-Switched Networks with Common Channel Signalling. *Proc. Australian Telecommunications and Applications Conference*, Melbourne, Australia (Dec 1994) 707–712.

[2] S.A. Berezner and A.E. Krzesinski, Quasi-Reversible Multiclass Queues with Order Independent Departure Rates. To appear in *QUESTA* (1995).

[3] S.A. Berezner and A.E. Krzesinski, Order Independent Queues with Losses. *Research Report RW95-01* (Jan 1995), Department of Computer Science, University of Stellenbosch, 7600 Stellenbosch, South Africa.

[4] S. Crosby and A.E. Krzesinski, Product Form Solutions for Multiserver Centers with Concurrent Classes of Calls, *Performance Evaluation*, Vol 11 No 4 (1990) 265–281.

[5] F.P. Kelly, Blocking Probabilities in Large Circuit-switched Networks, *Advances in Applied Probability*, Vol 18 (1986) 473–505.

[6] A.E. Krzesinski and R. Schassberger, The Multiserver Center with Hierarchical Concurrency Constraints, *Probability in the Engineering And Informational Sciences*, Vol 6 (1992) 147–156.

[7] J-Y Le Boudec, [1986], A BCMP Extension to Multiserver Stations with Concurrent Classes of Customers, *Proc. 1986 ACM Sigmetrics Conference, Performance Evaluation Review*, Vol 14 No 1 78–91.

[8] K.W. Ross and D. Tsang, Teletraffic Engineering for Product-Form Circuit-Switching Networks, *Advances in Applied Probability*, Vol 22 (1990) 657–675.